Intelligent Interface Learning with Uncertainty

Midwest Artificial Intelligence and Cognitive Science (MAICS) Conference 97, Dayton, OH, 1997

Robert A. Harrington and Scott M. Brown

Department of Electrical & Computer Engineering
Air Force Institute of Technology
2950 P Street, Wright-Patterson AFB, OH 45433-7765

Intelligent Interface Learning with Uncertainty

Robert A. Harrington and **Scott M. Brown**
Department of Electrical and Computer Engineering
Air Force Institute of Technology
Wright-Patterson AFB, OH 45433-7765
harringtonr@ssg.gunter.af.mil
sbrown@afit.af.mil

Abstract

This paper presents an intelligent user interface agent architecture based on Bayesian networks. Using a Bayesian network knowledge representation not only dynamically captures and models user behavior, but it also dynamically captures and models uncertainty in the interface's reasoning process. Bayesian networks' sound semantics and mathematical basis enhances it's ability to make correct, intelligent inferences as to the user's needs. We show explicit examples of our agent's reasoning using our Bayesian network and present results showing the utility of Bayesian networks in the domain of user interfaces.

Content Areas: user interfaces, agent architecture, cognitive reasoning, expert systems, reinforcement learning, probabilistic reasoning

Introduction

GESIA[1] (Generic Expert System Intelligent Assistant) is an intelligent user interface agent architecture conceived out of the development of a generic expert system shell (Harrington, Banks, & Santos 1996). This expert system shell, called PESKI (Probabilities, Expert Systems, Knowledge, and Inference) (Santos 1993), is a collection of expert system tools under one architecture that is designed to be totally independent of any application domain. The tools contained in PESKI include an inference engine (IE), a knowledge base (called a Bayesian knowledge base (Banks 1995; Santos & Santos 1996) or BKB), knowledge acquisition (KA) and associated edit supports (ES) tools, a knowledge base verification and validation (VV) tool, and a data mining (DM) tool. PESKI provides three types of communcation modes to the user — structured text, graphical, and natural language. This allows the user to use PESKI in a way most intuitive. Figure 1 shows PESKI's inference engine tool using the structured text communication mode. For more information on PESKI, see the United States Air Force Institute of Technology's Artificial Intelligence Laboratory web site (`http://www.afit.af.mil/Schools/EN/AI/`).

Intelligent user interface research is primarily focused on human-computer interface issues, especially with the abilities and usability of interfaces. However, intelligent interface researchers have put little emphasis on improving the structures representing the intelligence of these interfaces. In this paper, we are primarily concerned with presenting the utilization of Bayesian networks in intelligent user interfaces.

It is widely agreed that basing decisions on an accurate cognitive model of the user is important for effective prediction of user intent and that the interface should be able to collect and model information about false inferences (Oppermann 1994; Thomas 1993). Collecting such data is cognitively and computationally difficult (Hewitt & Halford 1993; Kuhme 1993).

Many research interfaces use rule-based intelligence (Gonzalez & Dankel 1993; Thomas 1993). Rule-based representations, as well as other knowledge representations (e.g., memory-based reasoning (Maes 1994)), fail in two key areas - representing uncertainty and dynamic user modeling. The use of "probability modules" (Winston 1984) is an ad hoc approach to determining answer reliability, i.e., uncertainty. Furthermore, the addition and deletion of rules to dynamically model a user is ad hoc. Therefore, knowledge representations that can dynamically capture and model uncertainty in human-computer interaction can improve the modeling of the user and user interface states in an intelligent user interface. One knowledge representation that is ideal for representing uncertainty is a Bayesian Network (BN). A Bayesian network is a mathematically correct and semantically sound model for representing uncertainty that provides a means to show probabilistic relationships between items (Pearl 1988).

[1]This research was supported in part by AFOSR #940006.

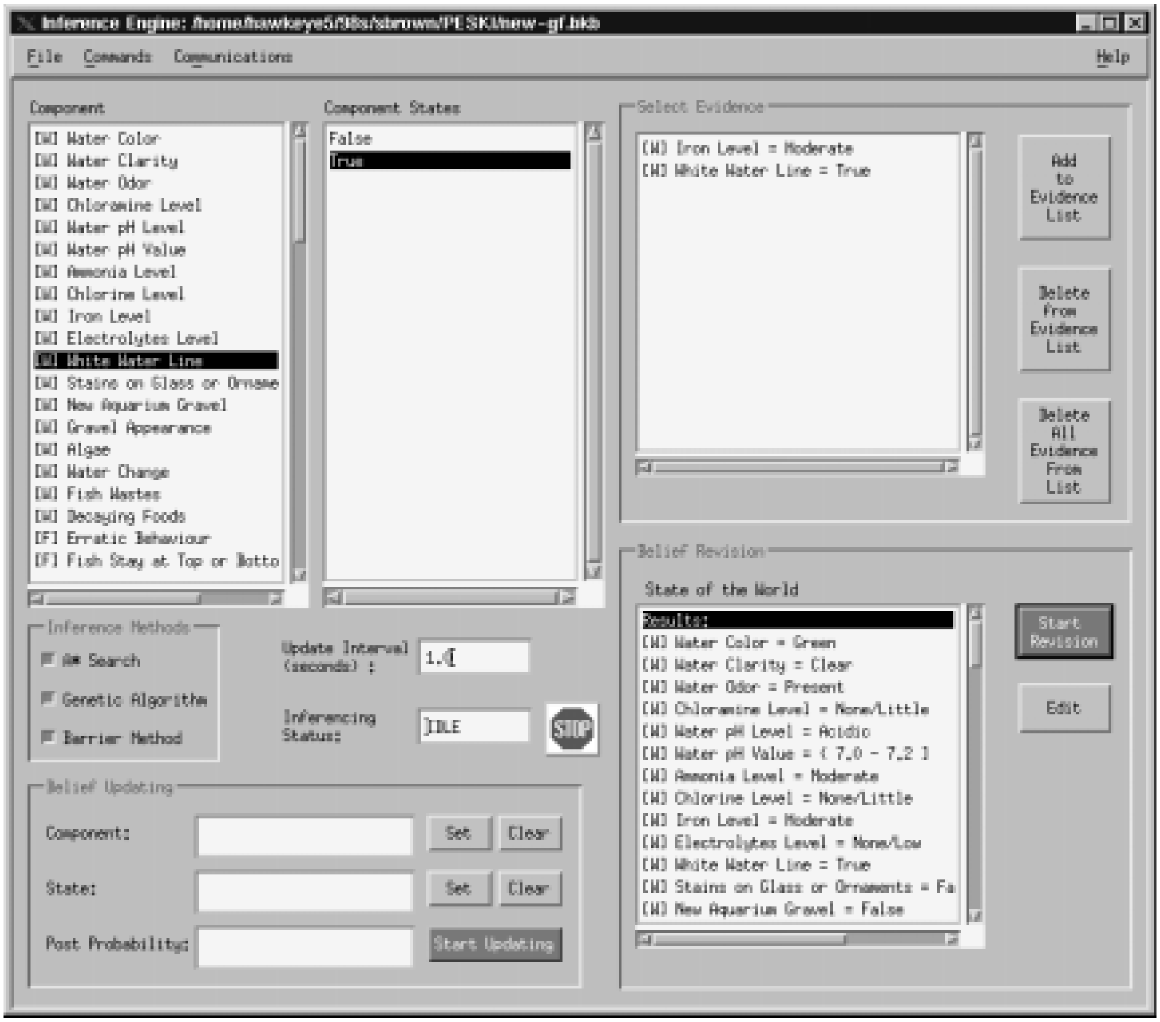

Figure 1: PESKI's Inference Engine Tool in Structured Text Communication Mode

GESIA Development

The goals of GESIA's development are threefold:

- To provide for user access to the many tools of the expert system using proven graphical interface design theory and implementation methods.
- To maintain the domain independence of the expert system, or, in other words, ensure the expert system can easily be used in multiple application domains.
- To assist the user with managing the complexities of the generic expert system through the use of intelligence or reasoning capability.

GESIA's user interface provides access to the generic expert system tools and applications. Figure 2 shows the three major layers of the architecture: the graphical layer, the intelligent interface agent layer, and the system layer.

The graphical layer of the architecture contains Motif/OSF standard interface widgets. Together, these widgets form the visual part of the interaction between the user and the expert system applications. The system layer provides the link between the GESIA's user interface and the expert system applications through a series of tool drivers, one for each expert system application tool.

The intelligent interface agent layer is the most complex and important layer of GESIA. This layer controls the communications and intelligence aspects of the interface and is composed of three layers: the adaptation layer, the adaptive layer, and the communications layer. The adaptation layer manages and tracks all adaptations the user makes to the user interface. The adaptive layer communicates directly with the interface learning network gateway (explained in the next section) to perform interface initiated adaptations to

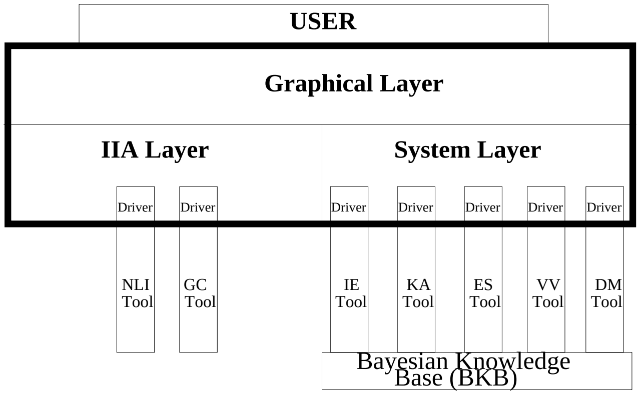

Figure 2: Layered Architecture of GESIA

GESIA's interface based on perceived user behavior. Finally, the communications layer controls the various modes of communication available to the interface such as structured text — provided via the standard X Windows list boxes, text boxes, etc., — graphical communication (GC) — provided via daVinci [2] as a means to graphically interact with the knowledge base, — and natural language interpreter (NLI).

The GESIA Interface Learning Network

The GESIA Interface Learning Network (hereafter referred to as the learning network) is the heart of the intelligent interface agent layer. The Bayesian network knowledge representation captures, stores, and models user and interface behavior. The network is composed of two semantically different nodes: interface learning nodes and interface information nodes. The network is also composed of containers that store learned user and user class behavior data and a network communications gateway.

Interface Learning Node

Semantically, the interface learning node represents behavior the interface has collected about a particular system user or class of users. This node is named according to the behavior collected, for example "User Prefers Knowledge Acquisition" or "User's Class Prefers Knowledge Acquisition." Each node's probability is stored as a fraction. The denominator of the fraction represents the number of learning occurrences that affect the node. The numerator of the fraction represents the number of learning occurrences that add to the truthfulness of the node (i.e., a higher probability).

[2]For specific information concerning daVinci, see `http://www.informatik.uni-bremen.de/ davinci/`.

After the user logs into the system, the interface learning network associated with that particular user loads stored data about the current system user into the interface learning node. Whenever the system user exhibits behavior represented by the node, the interface will call the node's update method to record the behavior. The updating is based on simple reinforcement learning. In this case, the behavior is recorded by incrementing the numerator and denominator of the fraction. For example, let us say we have a network composed of two interface learning nodes, "User Prefers Knowledge Acquisition" and "User Prefers Data Mining." If a user selects knowledge acquisition, the network will update the numerator and denominator of "User Prefers Knowledge Acquisition" and only update the denominator for "User Prefers Data Mining." In this way, the network has learned a preference for one expert system application over another. When the current user exits the system, the user's current learning network is saved.

Interface Information Node

Semantically, the interface information node represents a possible user state. Each interface information node is supported by two or more interface learning nodes and zero or more interface information nodes. When an interface information node is instantiated, it receives and stores access information to all its child nodes. The node sits "dormant" until the interface queries it for its probability, in order to make inferences as to the user's future state (i.e., (*user intent*). When the node is queried, this node combines the probabilities of all the supporting nodes to determine the probability that state is true using Bayes Theorem (Pearl 1988). This value represents the probability that the node's state is true. This node is named after the state it represents, for example, "User is Using Graphical Communication."

User and User Class Containers

These two parts of the interface learning network are responsible for storing all the learned user and user class data respectively. The user container controls all system logins, allowing the creation and deletion of users as well as normal logins. The user class container maintains information on the four user classes currently supported by PESKI: application user, application expert, knowledge engineer, and computer scientist. When a new user requests access to the system, the user is prompted as to what user type they belong. The new user's learned behavior data is initiated to his/her user class's interface learning network. Thereafter, all behavior of the user will not only affect the user's personal behavior (i.e., learning network) but also the behavior of the user class.

Interface Learning Network Gateway

The gateway provides the communication link between the learning network and the graphical user interface. All communication from the graphical user interface to the learning network must pass through the gateway. In this way, the gateway promotes information hiding for GESIA and maintains its generic nature. That is, GESIA need not be solely tied to PESKI, but usable with any system requiring an intelligent assistant.

Example of Network Use

This example of a simple network demonstrates how the network learns and how the learned data can be used to predict a user's behavior. Figure 3 depicts the network used in this example. This network has an interface information node, "User is Using Knowledge Acquisition" (UKA), that is supported by the interface learning nodes "User's Class prefers Knowledge Acquisition" (CPKA) and "User prefers Knowledge Acquisition (UPKA)."

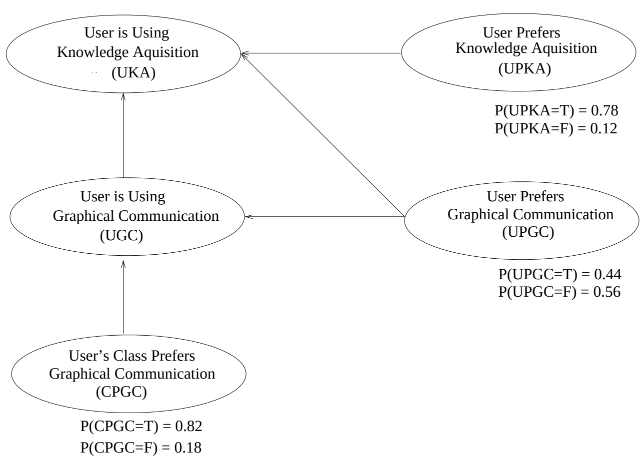

Figure 3: Simple Interface Learning Network

For this example, user JANE has logged onto PESKI through GESIA. The interface learning network loads all the learned data about JANE and sends the data to the appropriate interface learning nodes in the network, thus dynamically constructing the reasoning network. With the network loaded, JANE begins to use PESKI. As JANE performs actions through the interface, the interface records her behavior by calling the learning method of the IIA, which in turn updates the nodes related to JANE's behavior. For example, in Figure 3, if JANE chooses to use graphical communication from the communication mode menu of the interface, the interface will call the update data methods for the interface learning nodes CPGC (i.e., "User's Class prefers Graphical Communication") and UPGC (i.e., "User prefers Graphical Communication"). Thus, JANE's behavior is captured.

The usefulness of the network comes in predicting user intent (i.e., future behavior). For example, if the GESIA wants to predict what interface tool JANE will choose in order to automatically bring this tool up for her the interface will query the UKA (i.e., "Using Knowledge Acquisition") interface information node, calling the node's compute probability method. This method will then combine the probabilities of UKA's child nodes.

The probabilities are combined in the following way. First, we must construct conditional probability tables for the node of interest and all its parents' conditional probability tables, listing all possible combinations of the truthfulness of the causal parent nodes. The values for the conditional probability table are stored by the learning network as uncertainty supports.

The uncertainty supports for each node represent the uncertainty that the interface will make the correct choice when choosing a particular interface information node (interface state) as a future state. In other words, whenever the interface chooses a particular state as what the user will want next, the interface will store whether it was wrong or right about its choice. This value is used in the conditional probability table when the parent nodes are neither all true or all false. For Figure 3, the uncertainty support for the UKA node is 0.31. Therefore,

$$
\begin{array}{lllll}
P(UKA| & UPGC, & UGC, & UPKA) & = 1.00, \\
P(UKA| & \neg UPGC, & UGC, & UPKA) & = 0.31, \\
\ldots, & & & & \\
P(UKA| & \neg UPGC, & \neg UGC, & UPKA) & = 0.31, \\
P(UKA| & \neg UPGC, & \neg UGC, & \neg UPKA) & = 0.00.
\end{array}
$$

The uncertainty support for the UGC (i.e., "Using Graphical Communication) node is 0.65 and its conditional probability table is constructed similarly. Once the conditional probability tables are constructed, the probabilities may be combined using the chain rule (Pearl 1988):

$$
\begin{array}{ll}
P(UKA=T) = & P(UKA, CPGC, UPGC, UGC, UPKA) \\
 & +P(UKA, \neg CPGC, UPGC, UGC, UPKA) \\
 & \ldots, \\
 & +P(UKA, CPGC, \neg UPGC, \neg UGC, \neg UPKA) \\
 & +P(UKA, \neg CPGC, \neg UPGC, \neg UGC, \neg UPKA) \\
P(UKA=T) = & 1.00 * 0.82 * 0.44 * 0.65 * 0.78 \\
 & +0.31 * 0.18 * 0.44 * 0.65 * 0.78 \\
 & \ldots, \\
 & +0.31 * 0.82 * 0.56 * 0.35 * 0.12 \\
 & +0.00 * 0.18 * 0.56 * 0.35 * 0.12
\end{array}
$$

Therefore, P(UKA=T) = 0.5023 or 50%. Given this result, the user interface has utilized a mathematically sound method to capture user behavior and then convert it into a representation from which the user interface may reason about future user intent.

Implementation

The adaptive layer of the intelligent inteface agent layer are implemented in a prototype instantiation, providing basic adaptations such as prediction of tool use. Full implementation of communication mode layer have been designed for future completion. The communications layer of the intelligent assistant layer is currently implemented with structured text communication and graphical communication modes. The natural language communication mode is in development. Adaption of user interfaces (e.g., adding "hot keys" for often used actions) has been researched extensively. Therefore, research into the adaption layer has taken lesser priority and may be implemented in later revisions.

The interface learning network is fully implemented in C++. The network was tested by instantiating the 36 node network shown in Figure 4.

Testing the Agent

There are two basic types of testing performed for this research: prediction accuracy and usability. Prediction accuracy testing is performed to ensure the Bayesian network accurately captures user behavior. Usability testing explores the usefulness of the research product to real users.

Prediction Accuracy Testing

Prediction accuracy testing was accomplished by observing the dynamics of the agent's suggestion generation capabilities when given a set of test cases that mimic user behavior. We present the test case of the implemented system. Other cases can be found elsewhere (Harrington 1996).

As implemented, GESIA makes two main suggestion to the user upon login. The suggestions are a a combination of suggestions including Bayesian knowledge base (BKB) filename, system function, and communication mode. After system login, two BKB filenames are suggested. Once one is chosen or both rejected an additional two suggestions appear, giving suggested function and communication mode combinations. This combination or *double* suggestion must be completely true (both parts) for the user to accept it.

In this case the probabilities of the six interface information nodes used to determine what suggestions GESIA will make are tracked: "Using Knowledge Acquisition (UKA)", "Using Inference Engine (UIE)", "Using Text Communication (UTC)", "Using Graphical Communication (UGC)", "Using Full5.bkb (UFB)", and "Using Afit.bkb (UAB)". The user starts PESKI for the first ten times with the intention of using the Inference Engine function with Text Communication and Full5.bkb. This user accepts any suggestion that is completely true and rejects any suggestion that is not completely true. After the initial ten times the user switches their preference to the Knowledge Acquisition function with Graphical Communication and Afit.bkb, accepting or suggesting behavior based on these new preferences.

The results of this test are shown in Figure 5. These results clearly show the IIA's ability to quickly adapt to the user's change in preferences. It should also be noted the acceptance and rejection of suggestions, especially the rejection of suggestions that are partially but not fully correct, have an interesting affect on the probability distribution throughout the network. We "penalize" both parts of a rejected double suggestion, even when one part may be true. This fact is the reason for UAB's rapid increase in probability after the 17th step while UKA only maintains a steady rise in probability.

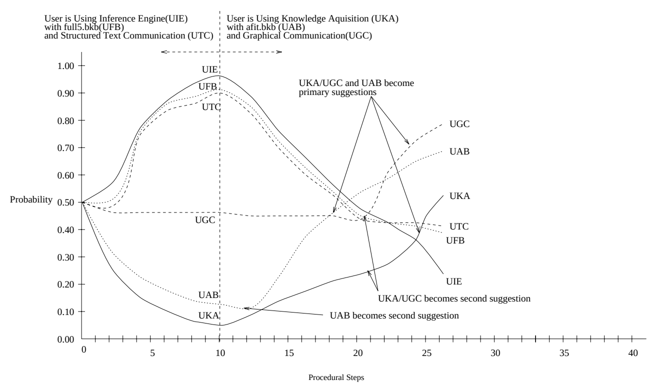

Figure 5: learning network Learning with Double Suggestions

Usability Testing

We have performed preliminary usability tests using GESIA with PESKI (Banks *et al.* 1997). There are three general tests we used to evaluate the reliability of interface intelligence. The first test is a collection of physical work requirements that quantify procedures the user must follow to get work done. How positively or negatively the user feels about using the interface intelligence is captured in the second test. The third test measures responsiveness burdens the intelligence places on the interface.

Physical Work Requirements Collecting the physical work a user performs is one way to evaluate the usefulness of the interface intelligence. Physical work requirements such as keystrokes, menu selections, reading, and button presses are collected for a user utilizing the interface intelligence. Care must be taken

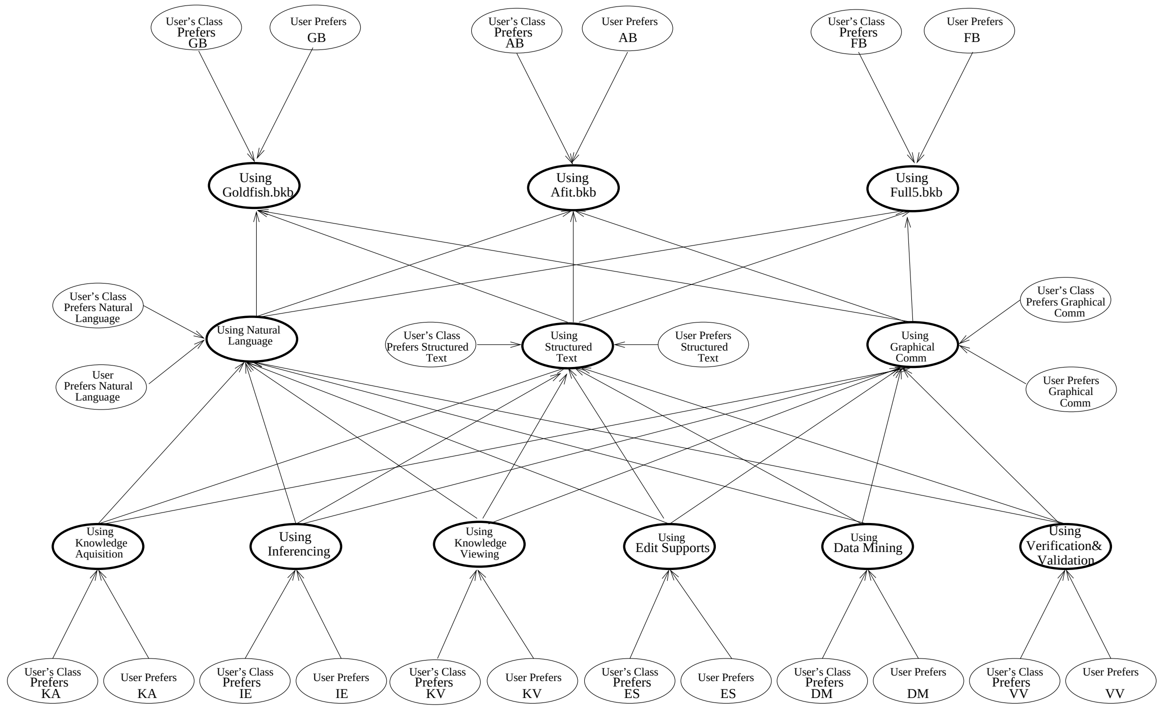

Figure 4: An Interface Learning Network for PESKI

when drawing conclusions from physical work requirements since this data does not form a complete picture of interface usability.

The current implementation of GESIA makes suggestions pertaining to what system function, communication mode, and BKB file the user wants to access at system startup. Therefore, this test concentrates on physical work required of the user if the user starts these choices themselves versus the physical work required if the user interacts with GESIA to make these choices.

Our results clearly showed using the GESIA's suggestions yields a considerable savings in physical work for the user. The physical workload without GESIA is seven-fold over that with the IIA. The data was compared with the data from prediction accuracy tests to ensure a true cost savings in physical work over time. The results show that over time the user receives a substantial work savings when using the IIA instead of making startup choices manually.

Acceptance of the Agent User acceptance data is collected by exposing a number of users to the interface intelligence and eliciting user opinion on a written survey. This survey is a carefully constructed list of instructions and questions that guides the user though GESIA's capabilities and requires exact and free-form responses from the user concerning these capabilities.

Users were generally satisfied with the timeliness of operations, although they seem to find the automatic operations performed by GESIA slightly slower than performing the same operations manually. This perception is a result of the specific implementation of GESIA discussed below.

Furthermore, users found the double suggestion of system function and communication mode more confusing than manually choosing the system function and communication mode from the main window. This result is somewhat supported by the fact that the work requirements for manually selecting the system function and communication mode are low.

The usefulness of GESIA's suggestions are as expected. Users generally found the IIA to be useful, although these results are most probably influenced by the results for user opinion on timeliness and complexity. A more indepth user acceptance study is desired to collect long term opinions of GESIA from many users using PESKI to perform real tasks.

Responsiveness of the Agent Responsiveness of an interface is typically an important criteria for interface users. Therefore, testing the responsiveness of the interface, particularly the effect intelligence has on interface responsiveness, is a collection of user opinions and empirical data. The user opinions are collect in the same manner described in the user acceptance test description. Empirical data is taken by collecting real time data during interface functions that are influenced by the interface's intelligent structures. Together, this data can give a good picture as to the acceptability of the intelligent user interface's responsiveness.

The initial implementation of Bayesian networks for use by the interface leanring network was not optimal. The real time data indicated the implementation of the IIA created some user noticeable pauses. The pauses were created mainly by the inferencing method used for the Bayesian network. The user acceptance study showed that users found the pauses noticeable but acceptable. Based on this feedback, we have reimplemented the Baysian networks and have greatly improved efficiency. Previous delays of three to five seconds have been reduced to under a second.

Future Research

The prototype interface learning network shown in Figure 4 needs to be expanded to capture additional actions. Expansion of the number of actions GESIA will monitor may allow for a more accurate model of the user's behavior. More importantly, though, is determining what is important to monitor in the domain to effectively and efficiently predict the user's intent. Research performed at Microsoft for their current implementation of *wizards* in Office 97 indicates they spent many hours determining how best to construct the Bayesian networks used (Horvitz 1996). Most researchers do not have the time nor the resources to perform such extensive knowledge acquisition. Therefore, we need to determine what is important for use in the system as the user is using the system, yet be able to make accurate predictions with limited knowledge. Research has begun in this area (Brown *et al.* 1997).

Our current implementation of GESIA uses a dynamic "hand-coded" interface learning network. We determine *a priori* the actions we will monitor. This is not unlike Maes' hand-coded situations (Maes 1994). This *a priori* determination limits the number of user actions we must monitor in our system. While although most "hand-coded" user models are static, ours allows the dynamic addition and deletion of nodes associated with a particular Bayesian knowledge base. We limit the number of BKB associated nodes allowed in the user's interface learning network at any one time. If the user loads a BKB that is not represented in the current network, we add it to the network. If we have reached our network size limitation (currently set at a hard limit of five BKB nodes), we delete the lowest probability node from the network. In this way, the most relevant (i.e., highest probability) nodes are in our network at any point in time.

The intelligence of the interface can be enhanced if the interface is able to interpret why it makes bad predictions. We propose a dynamic "meta-level" of inferencing, capable of modifying the user's interface learning network topology as the user performs action in PESKI. To realize this, we must be able to determine "real-time" what is happening with the user. Most usability studies are done "off-line" and have no immediate bearing on the user model. We desire to define several objective metrics (versus subjective usability comparisons) that give us insight into the accuracy of our user model and allow the network to be *dynamically* altered. The incorporation of temporal reasoning into this representation would allow the interface to predict user traits based on the patterns (Young & Santos 1996).

Conclusions

We have presented a new domain for the use of Bayesian networks. The interface learning network provides GESIA with an effective knowledge representation for user, user class, and interface behavior. The use of Bayesian networks over rule-based systems to accurately model the user better captures the uncertainty of user actions by using sound semantics and a firm mathematical basis. Initial tests show noticable savings in the user's physical workload while accurately predicting users' behavior. Furthermore, the momentum of learned behavior in one direction can be reversed and changed to another direction of behavior quickly.

Large networks will cause an exponential explosion in computations. This can reduce the practical size of a network using this representation. However, this can be overcome by placing simple restrictions on network topology. Furthermore, the learning network is only effective if used properly. In our research, the way suggestions were presented to users had a great impact on the users' evaluation of usefulness. Further usability studies will need to be conducted in order to determine the best way to present suggestions to the user.

References

Banks, S. B.; Harrington, R. A.; Santos, Jr., E.; and Brown, S. M. 1997. Usability testing of an intelligent intreface agent. In *Proceedings of Interfaces 97*.

Banks, D. O. 1995. Acquiring consistent knowledge for bayesian forests. Master's thesis, Graduate School of Engineering, Air Force Institute of Technology, Wright-Patterson AFB OH.

Brown, S. M.; Harrington, R. A.; Santos, Jr., E.; and Banks, S. B. 1997. User models, intelligent interface agents, and expert systems. In *Proceedings of Interfaces 97*.

Gonzalez, A. J., and Dankel, D. D. 1993. *The Engineering of Knowledge-Based Systems*. Englewood Cliffs, NJ: Prentice Hall.

Harrington, R. A.; Banks, S.; and Santos, Jr., E. 1996. Development of an intelligent user interface for a generic expert system. In Gasser, M., ed., *Online Proceedings of the 1996 Midwest Artificial Intelligence and Cognitive Science Conference*. URL: http://www.cs.indiana.edu/event/maics96/ Proceedings/harrington.html.

Harrington, R. A. 1996. Utilizing bayesian techniques for user interface intelligence. Master's thesis, Air Force Institute of Technology.

Hewitt, J. A., and Halford, P. G. R. 1993. Design of an intelligent interface to standard pc applications which maximizes the ability of the disabled user. *Knowledge-Based Systems* 6:24–29.

Horvitz, E. 1996. Lumiere project. World Wide Web Page. http://www.research.microsoft.com/ research/dtg/horvitz/lum.htm.

Kuhme, T. 1993. User-centered approach to adaptive interfaces. *Knowledge-Based Systems* 6(4):239–248.

Maes, P. 1994. Agents that reduce work and information overload. *Communications of the ACM* 37(7):811–821.

Oppermann, R. 1994. Adaptively supported adaptivity. *International Journal of Human-Computer Studies* 40:455–472.

Pearl, J. 1988. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. San Mateo, CA: Morgan Kaufmann.

Santos, Jr., E., and Santos, E. S. 1996. Representing and reasoning with bayesian knowledge-bases. *submitted to Journal of the ACM*.

Santos, Jr., E. 1993. A fully integrated probabilistic framework for expert system development. Technical report, Department of Electrical and Computer Engineering, Air Force Institute of Technology, Wright-Patterson AFB, OH.

Thomas, C. G. 1993. Design, implementation and evaluation of an adaptive user interface. *Knowledge-Based Systems* 6:230–238.

Winston, P. H. 1984. *Artificial Intelligence, 2d Ed.* Addison-Wesley.

Young, J. D., and Santos, Jr., E. 1996. Introduction to temporal bayesian networks. In Gasser, M., ed., *Online Proceedings of the 1996 Midwest Artificial Intelligence and Cognitive Science Conference*. URL: http://www.cs.indiana.edu/event/maics96/ Proceedings/Young/maics-96.html.